Conditional Random Field

Diego Marcheggiani

Corso di Elaborazione del Linguaggio Naturale

Pisa, May, 2011

Outline

Outline

Introduction

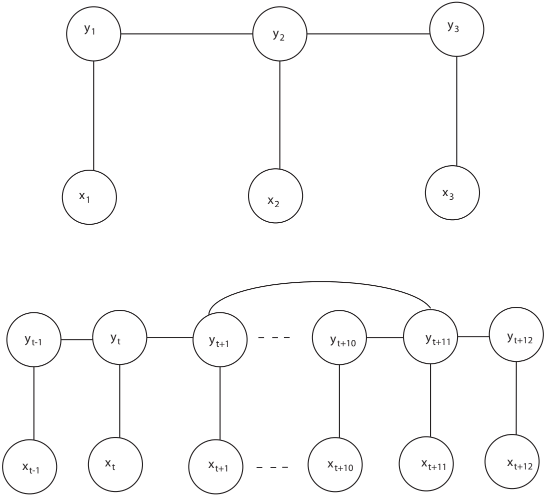
- Graphical probabilistic model
- Discriminative model
- State of the art in task as sequence labeling

Graphical Models

A graphical model is a family of probability distributions that factorize according to an underlying graph.

- They provide a simple way to visualize the structure of a probabilistic model.
- Inspecting such graph we can insight into the property of the model.
- Three equivalent types of model.
-
  - Directed graphical model (Bayesian networks)
  - Undirected graphical model (Markov random chain)
  - Factor graph (generalization of first two)

Graphical Models cont'd



Generative Models

Ng and Jordan, 2002

- Generative classifiers learn a model of the joint probability, $p(x, y)$, of the inputs x and the label y, and make their prediction by using the Bayes rule to calculate $p(y|x)$, and then picking the most likely label y.
- It's very hard to model the dependences between different features over the observed variables.
- In Naive Bayes classifiers we assume the conditional independence between features.

Discriminative Models

- Discriminative classifiers model the posterior $p(y|x)$ directly.

- Do not need to model the distribution (of features) over observed variables.

Outline

Linear-Chain CRF

Lafferty et al., 2001

$$P(\mathbf{y}|\mathbf{x}:\theta) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \theta_k f_k(y_{t-1}, y_t; \mathbf{x}_t)\right)$$

- T is the number of tokens in the sequence.
- K is the number of features we use in our model.
- θ is a parameters vector we have to estimate in order to obtain a model that fits the data.
- $Z(\mathbf{x})$ is an instance-specific normalization function.
- $$Z(\mathbf{x}) = \sum_{\mathbf{y}} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \theta_k f_k(y_{t-1}, y_t; \mathbf{x}_t)\right)$$

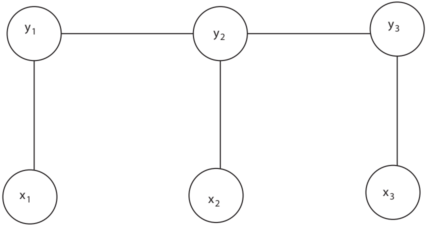
Feature Functions

$$P(\mathbf{y}|\mathbf{x}:\theta) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \theta_k f_k(y_{t-1}, y_t; \mathbf{x}_t)\right)$$

- f_k is one of the k feature functions
- $f_{ij}(y, y', x_t) = \mathbf{1}_{\{y=i\}}\mathbf{1}_{\{y'=j\}}$ for each transition from the state i to the state j.
- $f_{io}(y, y', x_t) = \mathbf{1}_{\{y=i\}}\mathbf{1}_{\{x=o\}}$ for each state-observation pair i, o.
- $\mathbf{1}_{\{y=i\}}$ is a function which returns $\mathbf{1}$ if $y = i$, 0 elsewere.

$\mathbf{x}_t$ is the feature vector at time t, it contains all the components that are needed for computing features at time t.
If we want to use x_{t+1} (the next word) as a feature, then the feature vector $\mathbf{x}_t$ is assumed to include x_{t+1}.

Feature Functions cont



The graph above is a classical Linear-Chain CRF.
We can modify and improve this model as we want.
For example:

$$f_{ij}(y, y', x_t) = \mathbf{1}_{\{y=i\}}\mathbf{1}_{\{y'=j\}}\mathbf{1}_{\{x=o\}}$$

e.g.

$$f(y_i, y_{i-1}, x_i) = \begin{cases} 1 & \text{if } x_i = John, y_{i-1} = O, y_i = NN \\ 0 & \text{otherwise} \end{cases}$$

Training

To estimate the θ parameters we calculate the maximum conditional log likelihood.

$$\ell(\theta) = \sum_{i=1}^{N} \log p(\mathbf{x}^{(i)} | \mathbf{y}^{(i)})$$

substituting the CRF model into the likelihood and adding a regularization parameter $\frac{1}{2\sigma^2}$ we obtain:

$$\ell(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{N} \log Z(\mathbf{x}^{(i)}) - \sum_{k=1}^{K} \frac{\theta_k^2}{2\sigma^2}$$

In general this function cannot be maximized in closed form. The partial derivatives are:

$$\frac{\delta \ell}{\delta \theta_k} = \sum_{i=1}^{N} \sum_{t=1}^{T} f_k(y_t^{(i)}, y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{y,y'} f_k(y, y', \mathbf{x}_t^{(i)}) p(y, y' | \mathbf{x}^{(i)}) - \sum_{k=1}^{K} \frac{\theta_k^2}{2\sigma^2}$$

Training cont'd

The function $\ell(\theta)$ is concave, that is every local optimum is a global optimum.

The maximization of the log likelihood can be computed with several numerical algorithms

- Gradient method of steepest ascent
- Iterative scaling
- Newton's method
- quasi-Newton's method BFGS and limited-memory BFGS

Forward-Backward Algorithm

In order to resolve two main problems in the CRF scenario, the $Z(\mathbf{x})$ evaluation and the computation of marginals distributions in the gradient computation, we have to use two algorithms belonging to the same class of algorithm, the forward-backward class.

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K}\theta_k f_k(y_{t-1}, y_t; \mathbf{x}_t)\right)$$

we define a function $\mathbf{g}$ as:

$$\mathbf{g}_t(y_{t-1}, y_t) = \sum_{k=1}^{K}\theta_k f_k(y_{t-1}, y_t; \mathbf{x}_t)$$

given that θ and $\mathbf{x}$ are fixed we can give as parameters y_{t-1} and y_t.

Forward-Backward Algorithm cont'd

Wallach, 2004

- $$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{t=1}^{T} \exp\left(\mathbf{g}_t(y_{t-1}, y_t)\right)$$
- for each t from 1 to $|T|+1$ we define an $|\mathcal{Y}|+2 \times |\mathcal{Y}|+2$ matrix called Transition Matrix:

$$M_t(u, v) = \exp \mathbf{g}_t(u, v)$$

- $M_1(u, v)$ is defined only for $u = START$, and $M_{|T|+1}(u, v)$ is defined only for $v = END$
- given this the only thing we have to do is multiplying all the matrices, e.g. $M_1 M_2, M_{1,2} M_3, \ldots$ and take the $(START, END)$ entry of the obtained matrix.

Forward-Backward Algorithm cont'd

In order to infer the marginal in the gradient computation, we have to use the forward-backward algorithm as in the HMM.
We have to solve the expectation of f_k under the model distribution

$$\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{y,y'} f_k(y, y', \mathbf{x}_t^{(i)}) p(y, y'|\mathbf{x}^{(i)})$$

Base case:

$$\alpha_0(y|\mathbf{x}) = \begin{cases} 1, & \text{if y = START} \\ 0, & \text{otherwise} \end{cases}$$

$$\beta_{|T|+1}(y|\mathbf{x}) = \begin{cases} 1, & \text{if y = END} \\ 0, & \text{otherwise} \end{cases}$$

Forward-Backward Algorithm cont'd

Recurrence relation:

$$\alpha_t(\mathbf{x})^T = \alpha_{t-1}(\mathbf{x})^T M_t(\mathbf{x})$$

$$\beta_t(\mathbf{x}) = M_{t+1}(\mathbf{x})\beta_{t+1}(\mathbf{x})$$

we can write:

$$p(y, y'|\mathbf{x}^{(i)}) = \frac{\alpha_{t-1}(y'|\mathbf{x}) M_t(y', y|\mathbf{x}) \beta_t(y|\mathbf{x})}{Z(\mathbf{x})}$$

Viterbi Algorithm

To determine the most probable sequence of labels $y_1, ..., y_T$ we use the Viterbi algorithm

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \sum_{t=1}^{T} \mathbf{g}_t(y_{t-1}, y_t)$$

We define:
$SCORE(y_1, ..., y_p) = \sum_{t=1}^{p} \mathbf{g}_t(y_{t-1}, y_t)$
$U(p) =$score best sequence $y_1, ..., y_p$
$U(p, v) =$score best sequence $y_1, ..., y_p$, where $y_p = v$

Viterbi Algorithm cont'd

Formally:

$$U(p, v) = \max_{y_1,\dots,y_{p-1}} [\sum_{t=1}^{p-1} \mathbf{g}_t(y_{t-1}, y_t) + \mathbf{g}_p(y_{p-1}, v)]$$

Base case:

$$U(0, v) = \begin{cases} 0, & \text{if } v = \text{START} \\ -\infty, & \text{otherwise} \end{cases}$$

we can recursively proceed in this manner:

$$U(p, v) = \max_{y_{p-1}} [U(p-1, y_{p-1}) + \mathbf{g}_p(y_{p-1}, v)] \forall v$$

Our goal is to find $U(|T| + 1, END)$

Outline

General CRF

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\mathbf{x}_c, \mathbf{y}_c; \theta_p)$$

where each factor is parametrized as:

$$\Psi_c(\mathbf{x}_c, \mathbf{y}_c; \theta_p) = \exp\left(\sum_{k=1}^{K(p)} \theta_{pk} f_{pk}(\mathbf{x}_c, \mathbf{y}_c)\right)$$

$$Z(x) = \sum_{\mathbf{y}} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\mathbf{x}_c, \mathbf{y}_c; \theta_p)$$

$\mathcal{C} = C_1, ..., C_P$ where each C_p is a clique template whose parameters are tied.

Outline

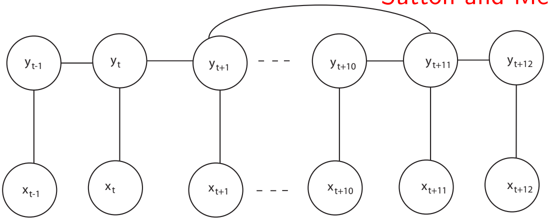
Important Tasks

As a versatile learning system we can employ it in several natural languages task:

- Sequence labeling, Part Of Speech (linear-chain)
- Information extraction, Named Entity Recognition (skip-chain, semi-markov)
- Text classification (multilabel crf)

Skip-Chain CRF

Sutton and McCallum, 2004



$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, \mathbf{x}) \prod_{(u,v)\in\mathcal{I}} \Psi_{uv}(y_u, y_v, \mathbf{x})$$

$$\Psi_t(y_t, y_{t-1}, \mathbf{x}) = \exp\left(\sum_k \theta_{1k} f_{1k}(y_t, y_{t-1}, \mathbf{x}_t)\right)$$

$$\Psi_{uv}(y_u, y_v, \mathbf{x}) = \exp\left(\sum_k \theta_{2k} f_{2k}(y_u, y_v, \mathbf{x}, u, v)\right)$$

Semi-Markov CRF

Sarawagi and Cohen, 2005

$$p(\mathbf{s}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{k}^{K} \sum_{j}^{|s|} \theta_k g_k(y_j, y_{j-1}, \mathbf{x}, t_j, u_j)\right)$$

- $\mathbf{s} = (s_1 ... s_p)$ denote the segmentation of $\mathbf{x}$ where
- $s_j = (t_j, u_j, y_j)$ where t_j is a start position, u_j is an end position, and y_j is the label of that segment.

Semi-Markov CRF employes a modified version of Viterbi algorithm that take into account the segment model

Multilabel CRF

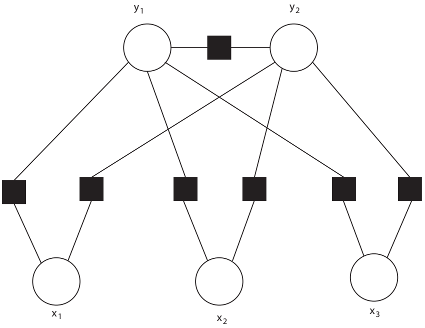
Ghamrawi and McCallum, 2005

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_k \theta_k f_k(\mathbf{x}, \mathbf{y}) + \sum_{k'} \theta_{k'} f_{k'}(\mathbf{y})\right)$$

$$k \in \{\langle v_i, y_j \rangle : 1 \leq i \leq |V|, 1 \leq j \leq |\mathcal{Y}|\}$$

$$k' \in \{\langle y_i, y_j, q \rangle : q \in \{0, 1, 2, 3\}, 1 \leq i, j \leq |\mathcal{Y}|\}$$

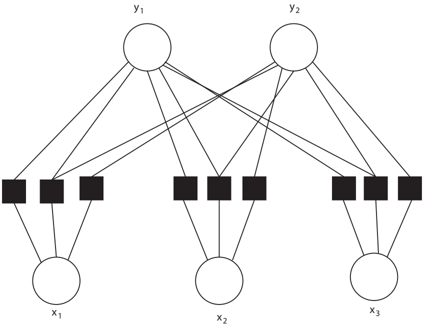
$\mathbf{y}$ = subset of the set $\mathcal{Y}$ represented by a vector of length $|\mathcal{Y}|$



Multilabel CRF cont'd

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_k \theta_k f_k(\mathbf{x}, \mathbf{y}) + \sum_{k'} \theta_{k'} f_{k'}(\mathbf{x}, \mathbf{y}) \right)$$

$$k \in \{\langle v_i, y_j \rangle : 1 \leq i \leq |V|, 1 \leq j \leq |\mathcal{Y}|\}$$
$$k' \in \{\langle v_i, y_j, y_{j'} \rangle : 1 \leq i \leq |V|, 1 \leq j, j' \leq |\mathcal{Y}|\}$$



Outline

Implementations

- CRF++ by Taku Kudo at http://crfpp.sourceforge.net/
- CRF Project by Sunita Sarawagi at http://crf.sourceforge.net/
- Stanford CRF at http://nlp.stanford.edu/software/CRF-NER.shtml
- Mallet at http://mallet.cs.umass.edu/index.php by Andrew McCallum

Outline

Conclusions

- CRF's take the best among HMM's and ME's
- State of the art in many NLP tasks
- A rich framework such as HMM's

Questions?

Thanks for your attention!